6D Object Localization and Obstacle Detection for Collision-Free Manipulation with a Mobile Service Robot

Jens Kuehnle
Alexander Verl
Fraunhofer IPA
Nobelstrasse 12
Stuttgart, Germany
jens.kuehnle@ipa.fraunhofer.de

Zhixing Xue
Steffen Ruehl
J. Marius Zoellner
Ruediger Dillmann
Forschungszentrum Informatik
Haid-und-Neu-Str. 10-14
Karlsruhe, Germany
xue@fzi.de

Thilo Grundmann
Robert Eidenberger
Raoul D. Zoellner
Siemens Corporate Technology
Information and Communications
Otto-Hahn-Ring 6
Munich, Germany
thilo.grundmann.ext@siemens.com

Abstract—The main goal of mobile service robots is to assist human beings in their everyday life. This implies that service robots will have to deal with human environments rather than with artificial structured lab areas. Thus, one of the most important steps towards able service robots is to enhance their ability to operate well in these unstructured environments. In this paper we present an approach to recognize and localize as well as safely manipulate objects in the presence of obstacles. Our method was implemented, has been tested and was presented on the 2009 CeBIT technology show in Hanover, Germany using the experimental robot of the DESIRE project.

Keywords: manipulation, 6D object localization, obstacle detection, stereo vision, 3D time-of-flight camera

I. INTRODUCTION

One of the main reasons that robotic assistance is not yet available for home use is the high variability of human living areas. State-of-the-art robots are comfortable with clearly structured environments as present for instance in production lines. Additionally these areas commonly have been adjusted gently to support robotic requirements. In contrast to that, no human user will accept major changes to his home only to meet the special needs of a service robot. Therefore, service robots can not expect clean structured areas but rather have to cope with unstructured ones full of unknown obstacles, bad lighting and other pitfalls.

In this paper we focus on the problem of unknown or unrecognized objects in the context of object manipulation. The problem is, that any object recognition method based on sensor data is erroneous to a certain degree and will provide false positives as well as false negatives at a certain rate. These rates depend on the specific recognition method, and it is challenging to find one single method that fulfills all the needs of the manipulation task. In a manipulation task we need to know the exact pose and class of the object that is to be manipulated, and we need to know where the arm and hand can move without any collision with other objects.

As we do not need to know what class these other objects belong to we consider them as obstacles.

To meet our requirements, two approaches with very different statistical properties are combined. The first one of object recognition, is able to detect and localize objects of classes that are known to the system based on a stereo camera system with a very low false positive rate, facing undetected objects due to lighting conditions or simply because they are not known to the system. The other one detects unclassified obstacles, based on an active 3D time-of-flight camera, with a very low false negative rate. In combination we achieve a system that is able to localize the objects and arbitrary obstacles in the scene, so that the known objects can be manipulated securely.

The paper is organized as follows: first we will discuss the state of the art regarding robot systems designed for manipulation tasks. In section III we will present our robot system. Section IV will describe in more detail the components that work together in a manipulation task. In section V we focus on the integration of the components presented in section IV into a fully functional manipulation subsystem. In section VI experimental results are presented.
II. STATE OF THE ART

There is a large number of systems known that examine a similar scenario to ours, so we will mention only a few exemplary systems. One of the earlier ones is the Mobman [1], a system equipped with a single arm and a simple gripper that was presented in 2002 and was able to grasp simple objects that needed to be located on a table. The ARMAP system [2] is a two-armed humanoid with manipulative, perceptive and communicative skills. The focus of the robot Justin [3] lies on the dexterous two-handed manipulation and control methods for the arms and hands. Khatib and Brock [4], [5] put a strong focus on mobile manipulation under hard dynamic constraints and planning in such dynamic scenarios. Other work related to planning of manipulation tasks are considering the manipulation tasks embedded in the interaction loop with humans [6].

Considerable research work has also been focused on the development of humanoid robots [7], [8] which are however mostly only equipped with simple and light gripping devices and have relative poor manipulation skills. Wide public attention has been obtained by the humanoids from Toyota, but the technical details of these systems are kept secret.

III. SERVICE ROBOT DESIRE

The work presented in this paper is realized on the experimental platform DESIRE. It is designed to execute typical task in everyday household environments such as bringing services. In order to be suited to daily use, one of its most important, safety-related features is the ability to manipulate objects in its environment collision-free on the basis of a comprehensive and integrated scene model containing not only the locations of recognized objects but also obstacle information within the robots’ workspace.

Figure 1 shows the experimental platform DESIRE. It consists of an omni-directional drive, a body, one sensor head, two arms and two hands. The sensor head is mounted on a pan-tilt-unit and is equipped with a stereo camera head, two arms and two hands. The sensor head is mounted consists of an omni-directional drive, a body, one sensor head, two arms and two hands. The sensor head is mounted.

A. Object Recognition and Localization

In order to achieve the required precision in full 6D object localization that is needed for robust robotic manipulation with a multi-fingered hand as done in [11], we use a stereo algorithm based on sift features, a state-of-the-art pattern recognition method [12]. The classic sift interest point consists of a 2D location, a scale, an orientation and a descriptor $s = \{u, v, s, \phi, d_l\}$. We however use an extended version of the interest point which consists of a 3D location, a scale, a viewpoint direction, an orientation and a 128 dimensional descriptor $s_{\#} = \{x, y, z, s', x', y', z', \phi, d_l\}, i \in \{1...128\}$.

In a preprocessing step, we construct a 3D sift model $M$ of each object to be considered. The data basis is given by the common raw data entities defined in DESIRE. The recognition and localization process consists of the following steps (simplified for the sake of clarity):

1) Calculate a set $S_i = \{s^1, ..., s^n\}$ of standard sift features in one image of the stereo pair
2) For all elements $s^i \in S_i$ try to find a match $c^i = \{i, j\}$ with a 3D feature from the model database $s_{\#}^j \in M$
3) Find subsets $S_{\#}^h \subset S_i$ with high consistent support for one hypothesis: object classification and rough 3D localization
4) For every subset $S_{\#}^h$ calculate corresponding set on the other image $\tilde{S}_{\#}^h$ by finding stereo correspondences using the descriptor, the epipolar and the scale constraint.
5) Construct 3D feature sets $\tilde{S}_{\#}^h$ via stereo triangulation for every pair $\{S_{\#}^h, \tilde{S}_{\#}^h\}$
6) Transform each 3D sift cloud $S_{\#}^h$ into the subset $M_{\#}^h$ of the 3D model sift cloud by using the correspondences $c^i$ from step 2: object localization
7) Transform the resulting 6D poses into the world coordinate system using pose of the sensor head

IV. COMPONENTS

In this work, we assume the objects to be manipulated to be known in advance. In consequence, all other objects must be thought of being unknown and thus, regarded as obstacles. Only known objects can be manipulated, though in such a way as to avoid collisions with the remaining scene (i.e. other known as well as unknown objects).

In the following subsections, we describe the key components required to achieve collision-free manipulation of objects. In particular, object recognition and localization of known objects (IV-A), obstacle modeling of unknown objects (IV-B), grasp and path planning (IV-C). The way in which these components work together to achieve collision-free manipulation is explained in section V.

All models used by these components must be consistent throughout the system. The models that are important in the scenario presented in this paper are either environment-related or object-related. The former are models that describe the robot’s environment (e.g. wall, door, table), and we assume those entities to be fixed in space. The latter are models of the objects that are supposed to be manipulated. An arbitrary number of those objects can appear in the robot’s environment. In DESIRE, all object models are based on an unique raw data set that is established with the Interactive Object Modeling system described in [10]. Each object considered constitutes a raw data entity with a 3D triangle mesh of its surface and 396 stereo images taken from different view points. Since the 3D triangle mesh and the stereo views are registered with respect to each other, an unique object coordinate reference frame is established during modeling.
After having calculated the $S_l$ that commonly consists of about 1,500 - 8,000 features, the correspondences $c^i_l$ with reference to the object model database $M$ have to be established. We use a kd-tree to perform an approximate matching of the descriptor that delivers multiple matching candidates $c^{ik}$ as the uniqueness of interest points can not be assured in the context of multiple object databases.

Each of these matchings $c^{ik}$ already constitute a hypothesis $h^{ik} = \{\text{class, pose}\}$ that consists of classification and 6D localization in our context. This is based on the fact that in our model database $M$ the position of an object with reference to the camera and its class are known. In combination with the 2D location, scale and orientation of the measured interest point $s^i$ a rough 6D localization is given.

Due to the nature of the sift matching a lot of these single hypotheses $h^{ik}$ are erroneous. Therefore we search for peaks in the seven dimensional hypothesis space and regard the members of these peaks as measurements that originate from the same object. The members of each peak constitute a subset $S^h_l$ as mentioned in step three.

At this point one can apply the standard method for pose estimation based on mono pictures, the POSIT [13] algorithm, yet we were not able to fulfill our high precision goals by doing so. To suit our requirements we developed a stereo approach that is naturally suited to increase particularly the low precision of the distance measurement which was the main drawback of POSIT in our scenario.

Our approach takes all subsets $S^h_l$ and searches for the corresponding features $S^h_R$ in the other camera’s image. The first rough localization result that we have already serves as input for a region-of-interest that increases the speed of the second sift calculation significantly. For each interest point that was found in both pictures a 3D position in the camera frame can be determined using standard triangulation methods, given a precise stereo calibration, leading to a 3D sift interested point as described in the beginning of this section.

The resulting set $S^h_l$ corresponds to a subset $M^h$ of the object’s 3D sift model $M$ with known matches $c^i$ between both models, hence a simple iterative closest point method delivers the 6D pose of the object in the camera frame. To account for mismatched interest points, the result of the ICP is evaluated regarding the distance of the interest point pairs, dropping the ones that have a higher distance than twice the mean distance. If this process drops pairs due to their errors the ICP is evaluated again. In the end a theoretical minimum of three pairs is required to find a pose, however experiments showed that high accuracy requires a minimum of six pairs.

One important advantage of this 3D sift stereo approach is, that in contrast to other approaches it is able to handle objects of any shape, as long as the object fulfills the requirements on its texture that are inherent to the sift algorithm. In a further stage it is planned to incorporate state of the art methods for non-textured object recognition to increase the range of object types the robot can handle.

B. Obstacle Modeling

In order to obtain surface data, we rely on a solid state 3D time-of-flight (TOF) camera that is capable of capturing range information at video frame rates. Instead of providing the illuminance (i.e., grey-scale or color) as conventional CCD/CMOS chips do, such a sensor measures the viewed scene with respect to distance. For this purpose, the sensor emits modulated light, which is reflected by objects in the scene and projected onto the chip. In each pixel, the reference signal is correlated with the modulated light that it receives. The distance of the scene along the line of sight of any pixel is half the distance the light beam covers in the corresponding direction. Since the distance is recovered in each pixel, the sensor yields the surface of the objects in the scene observable in the sensor’s viewpoint.

Since TOF cameras are still somewhat prototypical, there are numerous factors that affect the accuracy of distance measurements. Lots of research is being done to understand the effects and to compensate the errors introduced [14], [15], [16]. However, a complete model to correct all of them is still missing.

In order to get rid of the distortion of the lens and the misalignment of the chip, the sensor is calibrated using the MATLAB Camera Calibration Toolbox [17] based on 100 intensity images. By means of the estimated intrinsic and extrinsic camera parameters, a point cloud in Cartesian coordinates corresponding to the radial distance measurements of the depth map can be computed. However, the sensor possesses considerable variability in its depth sensing range. As a result, planar surfaces are perceived as being curved. As described in the literature [14], the effect can be corrected by means of a plane calibration. This offline correction is however only valid under the internal and external conditions present during calibration. Influencing conditions are for example temperature, ambient light, surface reflectance and angle of incidence. Our experiences with the sensor suggest that an online depth correction on each individual depth map is inevitable in order to obtain stable and reliable measure-
ments. We address this problem by using environment-related knowledge as nominal reference to fit the actual data to. In doing so, we estimate an optimal transformation between the reference scene and the corresponding measurement point cloud and correct the data accordingly. In the work presented, we use planar surfaces within the robot’s environment (such as walls, tables, etc.) and perform plane fitting. Of course, we have to account for occlusion of known surfaces by unknown ones. For this purpose, we complement the data fitting with a data segmentation step that partitions the measurements into inlier (i.e., data points belonging to the model feature to be fitted) and outlier (i.e., data points not belonging to it). As segmentation criterion, we use the Euclidean distance and assume the error of fit to be equal to an order of magnitude of the noise level of the measurement device. We keep in mind though that this depth correction procedure requires environment-related knowledge.

The distance measurements of TOF cameras experience considerable noise. Therefore, smoothing of the data is essential. Furthermore, jump edges (i.e., gaps in distance between neighboring pixels due to occlusion) are integrated over such that the gaps are filled with pseudo measurements. We process the sensor’s depth map with a combination of edge-preserving smoothing and jump edge-finding in order to filter the pixels corresponding to pseudo measurements.

The obstacle modeling uses a z-buffer filtering algorithm to reject apparent parts in the TOF camera’s field of view (e.g., recognized and localized objects, robot’s surface). It renders an artificial view of the such parts in the corresponding view point. This artificial view is used as a mask on the current measurements to find all pixels that do constitute actual obstacles. Obstacles are either approximated by bounding boxes or represented by 3D triangle meshes of the obstacle surface as well the occluded area.

C. Grasp and Path Planning

To grasp objects in everyday life environments with the Schunk Anthropomorphic Hand (SAHand), the grasp simulator ”GraspIt!” [18], [19] is used to plan high quality grasps. The method is based on the observation that humans unconsciously simplify the grasping task to selecting one of only a few different preshapes appropriate for the object and for the task to be performed. Such hand preshapes with finger postures for the SAHand are predefined. In the grasp simulation, the hand with one of the preshapes moves from the starting position relative to the object pose along an approach direction towards the object. The fingers close around the object. After the object is grasped, the contact points between the hand and the object are collected to evaluate the grasp quality. “Largest sphere in grasp wrench space” is used as grasp quality measurement, so that the grasp can resist with independence of the perturbation direction [20].

The method introduced in [18] needs a manual decomposition process to decompose the object geometric model into several shape primitives, such as sphere, box cylinder or cone, from which the approach directions are determined. Like the shape primitives, superquadrics can also be used to approximate the object shape. The decomposition trees [21] are used to create the approach directions from the point cloud of the object for the grasp planning. Main benefit of this method is that it does not need the manual decomposition process. So that after the object is scanned by a laser scanner, its geometric model and the approach directions can be automatically generated, as well as the grasps for this object.

By finger closing, it is desired to find the first contact point between the finger link and the object using the collision detection technique. This is a problem of continuous collision detection, which does not only check the collision between two static objects, but also takes into account the relative motion between them and finds the first time of contact. In [19], a Newton-Raphson method was used to find this contact point with the complexity of \(O(\log n)\). Conservative Advancement method was used in [22], which computes the distance between two convex parts and computes the further relative movement without any collision. In the next iteration step, the moving object is placed at the computed position, until the distance between them are smaller than a predefined distance threshold, with which the objects are treated as colliding and the contact point between them is found. The collision can be computed within only a few iteration. Its complexity is reduced to a constant value, \(O(1) \approx 2.1\) in our tests, with \(0.1\text{mm}\) as the distance threshold. With the efficient continuous collision detection technique, each grasp candidate can be computed within only \(50\text{msec}\).

After all the grasp candidates are tested in the simulation, high stable grasps are saved in a grasp database. A probabilistic collision free path planner [23] is used for motion control.

V. INTEGRATION

A. System Architecture

In a complex and distributed system with various subsystems each performing an individual task, the integration of all functionalities has to bridge the borders between hardware and software. The DESIRE project uses a custom-made CORBA-based middleware [24]. Each component provides its functionality through a service interface formulated in the interface definition language (IDL). Following the programming-by-contract principle, each service guarantees
to produce defined postconditions as long as specific preconditions are fulfilled. We distinguish two types of services: operations and commands. Operations are synchronous calls that block the calling process and directly provide the service’s results. Commandos are asynchronous calls that do not block but rather notify the calling process about the service’s progress and on completion provide means to retrieve the results indirectly.

The system is controlled by means of three components: planner, sequencer and status model. The planner makes decisions and computes plans to solve a given problem. The sequencer executes issued plans by calling a sequence of services and coordinating parameters. The status model manages the status of the system (e.g. status of each component, kinematic model of entire robot, robot-in-world pose, head-in-world pose) and conducts continued troubleshooting.

The wishlist concept provides a component the means to petition a change of the system’s status in order to optimize its own performance (e.g. a perception component can issue a look-at-world-point wish in order to try to identify objects on the table, the manipulation component can issue a drive-to-world-pose wish in order to reach an object on the opposite side of the table). The wishlist concept follows the idea that a component knows in which conditions it expects to perform best. Although the exact configuration of the system is unknown to the component, it can use a wishlist in order to try to establish ideal conditions. The sequencer decides on whether or not to grant a wish and dispatches it accordingly (e.g. a sensor-at-world-pose wish can cause the drive control to move the robot and the head control to change pan-tilt angles).

Using the described architecture, the components become self-contained entities that provide component-defined services. A component need not be aware of any particular aspects of other components or the exact configuration of the system. It just needs to conform with the interfaces defined.

In the following subsections, we describe the way in which the proposed system architecture can be used in order to achieve a comprehensive and integrated scene model for collision-free robotic manipulation. For the sake of clarity, we use a simplified illustration.

### B. Comprehensive and Integrated Scene Model

We establish a comprehensive and integrated scene model that consolidates information on known objects (i.e. recognized and localized as discussed in subsection IV-A) and unknown objects (i.e. the remaining objects in the scene being classified as obstacles as discussed in subsection IV-B). The mechanism is illustrated in the sequence diagram of Figure 4. Two aspects of the scene model are to be noted.

First due to adverse conditions, the object recognition can fail to localize objects although contained in the object model database (e.g. texture is not visible due to light reflection on the object’s surface). Since such objects cannot be manipulated, the obstacle modeling must succeed in classifying them as obstacles. Using the object modeling in a stand-alone fashion, all known and unknown objects as well as parts of the robot in the field of view of the TOF camera are identified as obstacles prohibiting any attempt to execute manipulation tasks whatsoever. In both situations, apparent parts of the environment are misclassified as obstacles. We surmount this problem by providing the object modeling algorithm the information contained within the current scene model as well as the all joint angles of the platform (i.e. arm joints and pan-tilt unit) in order allow for z-buffer filtering of the apparent parts. Obstacle modeling is performed on the remaining scene.

Second, the scene model can wish to adjust the pose of the head in order to enhance performance (e.g. position camera such that table is in field of view, reposition camera in order to see occluded parts). A detailed discussion on perception planning is given in [25].

### C. Collision-Free Manipulation

Collision-free manipulation of a specified object in the scene model is achieved by determining a feasible grasp in the grasp database that can be performed collision-free and by executing it. In doing so, not only the pick-operator but also the place-operator is checked for collision-free execution. First, the robot’s environment is represented by three sorts of data. Parts of the environment such as walls, doors and tables are considered static in the environment model. Second, the triangle meshes in the object database of the known objects recognized in the scene model are transformed and located in the environment model. Third, the triangle
meshes approximating unknown objects are considered dynamic in the environment model. In grasp determination as well as execution, the robot is not allowed to collide with the environment model except for the robotic fingers that are allowed to collide with the object to be manipulated at desired contact points. A probabilistic collision-free path planner \[23\] is used to bring the robotic arm to the desired pick position. As soon as the object is in the robotic hand, it is treated as part of the kinematic chain in the following place task in order to also avoid collisions between the object in the robotic hand and the environment model. Further manipulation of the object (e.g. regrasping) can be performed as well. The robotic arm is always operated in impedance control mode to comply with environment deviations. The mechanism is illustrated in the sequence diagram of Figure 6.

Two aspects of collision-free manipulation are to be noted. First, obstacle modeling notifies grasp and path planning in case changes are registered. If parts of the robot enter and move in the TOF camera’s field of view, then occluded features are maintained.

Second, the collision-free manipulation can wish to adjust the pose of the robot in order to enhance performance (e.g. reposition robot on the opposite side of the table).

VI. EXPERIMENTAL RESULTS

The introduced robot was presented on the 2009 CeBIT technology show in Hanover, Germany for one week. “Clean up the table” was the selected manipulation task for the event. Up to 10 typical objects in household were placed randomly on the table in front of the robot, which should be brought back to the two baskets autonomously by the robot. Only the table information is a priori known for the system, not the information about the objects. The robot should use its perception ability to find out which objects are on the table and where they are. The other system unknown objects can also be placed on the table, which will be treated as obstacles, which the robot may not collide with. To clean up the table, the robot plans autonomously, in which order the objects can be grasped and brought into the basket. In the whole week, more than 500 grasp operations were successfully performed for the public.

VII. CONCLUSION

Using the described mechanisms the DESIRE robot is able to perform collision-free manipulation in everyday life environments. It uses a comprehensive and integrated scene model that contains not only known objects but also models unknown ones in order to regard them as obstacles. Experiments show, that the components perform excellently and allow for robust and error-free collision-free robotic manipulation.

VIII. ACKNOWLEDGMENTS

This work was funded as part of the research project DESIRE by the German Federal Ministry of Education and Research (BMBF) under grant no. 01IME01A, 01IME01E and 01IME01D.

REFERENCES


